UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/618,277 | 07/11/2003 | Baskaran Dharmarajan | MS1-1565US | 4822 |

22971 7590 07/07/2008
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052-6399

| EXAMINER | |
|---|---|
| LE, MIRANDA | |
| ART UNIT | PAPER NUMBER |
| 2167 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 07/07/2008 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

roks@microsoft.com
ntovar@microsoft.com

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/618,277 | DHARMARAJAN ET AL. | |
| | Examiner | Art Unit | |
| | MIRANDA LE | 2167 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *17 March 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This communication is responsive to Amendment, filed 03/17/08.

Claims 1-26 are pending in this application. Claims 1, 7, 13, 18, 21, 24 are independent claims. In the Amendment, claims 27-28 have been added, and no claims have been amended, or cancelled. This action is made Final.

### *Claim Objections*

Claim 21 is objected to because of the following informalities: Claim 21 needs to include a processor or memory to execute the software components. Appropriate correction is required.

### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time

a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Goward et al. (US Pub No. 20020120677), in view of Viswanath et al. (US Pub No. 20030125966).

**As to claims 1, 18**, Goward teaches a method/apparatus comprising:

receiving a request for a Web page *(i.e. client sends request to web sever, step 402, Fig. 4; [0036; 0043-0050])*;

identifying an Active Server Page *(i.e. Web Server selects Servlet to execute, Step 404; Servlet selects Server Page to execute, Setp 410; retrieve server page form file, Step 502, Figs. 4, 5)* includes a complied user interface template created *(i.e. templates 201-203, Fig. 2; select a static HTML page, a server page or a graphics file to display, [0040]), (i.e. compile server page, Step 504, Fig. 5)* using an Active Server Page Language *(i.e. the server page includes an Active Server Page (ASP), [0019],* Fig. 6*)* which when compiled is executed through an application programming interface developed *(i.e. Application server, Fig. 1)* using a system language to generate the requested Web Page in the system language from the user interface template created using the Active Server Page Language *(i.e. execute complied server page to compute data and generate page, Step 506, Fig. 5; [0036; 0043-0050])*;

executing the Active Server Page through the application programming interface to generate the requested Web Page *(i.e. Servlet invokes method to execute server page, Step 412, Fig. 4; [0036; 0043-0050])*; and

providing the requested Web Page to a source of the request *(i.e. output display page, Step 510, Fig. 5; [0036; 0043-0050])*.

Although Goward teaches the server page is an Active Server Page (ASP), Goward does not explicitly disclose "Active Server Page Language".

Viswanath teaches the Active Server Page *(i.e. Active Server Pages (ASP) by Microsoft, Inc., along with or instead of an actual JSP, [0064]), and scripts.*

Both Goward and Viswanath teach the same field as Active Server Page as well as Java Server Page in a network environment. It would have been obvious to one of ordinary skill of the art having the teaching of Goward and Viswanath at the time the invention was made to modify the system of Goward to include the limitations as taught by Viswanath. One of ordinary skill in the art would be motivated to make this combination in order to compile JSP-S into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server in view of Viswanath ([0048]), as doing so would give the added benefit of providing a method and/or system for handling requests from a user and rendering a resulting page in HTML useful in effectuating aspects of an order management system, which reduces the redundancy in the underlying code required to effectuate the page as taught by Viswanath ([0009]).

**As to claims 7, 21**, Goward teaches a method/apparatus comprising:

identifying *(i.e. Web Server selects Servlet to execute, Step 404; Servlet selects Server Page to execute, Setp 410; retrieve server page form file, Step 502, Figs. 4, 5)* a plurality of user interface templates *(i.e. templates 201-203, Fig. 2; select a static HTML page, a server page or a graphics file to display, [0040])* created using an Active Server Page Language *(i.e. the server page includes an Active Server Page (ASP), [0019],* Fig. 6*)* and associated with a Web-based application *(i.e. server page, [0047])*;

compiling *(i.e. compile server page, Step 504, Fig. 5)* each of the plurality of user interface templates into a single file *(i.e. compile server page, Step 504, Fig. 5)* containing a plurality of byte codes *(i.e. the server page includes dynamically executable code embedded in a display language, [0011])*, wherein the byte codes are capable of being executed by an execution engine that implements an Internet service application programming interface (ISAPI) *(i.e. main program of Servlet 110, [0048])* of the Web-based application *(i.e. During this execution process, server page 211 can make callbacks into servlet 110, and also to use methods defined within servlet 110, and also to access state information stored within servlet 110, [0045])*; and

executing the plurality of byte codes when the Web-based application is executed *(i.e. Servlet invokes method to execute server page, Step 412, Fig. 4; [0036; 0043-0050])*,

Goward implicitly teaches Active Server Page Language as in Fig. 6 and in [0048] *(i.e. server page 211 includes HTML markup text, [0048])*.

Goward implicitly teaches byte codes as executable code *(i.e. the server page includes dynamically executable code embedded in a display language, [0011])*.

Goward does not explicitly teach "Active Server Page Language"; "byte codes".

Viswanath teaches the Active Server Page *(i.e. Active Server Pages (ASP) by Microsoft, Inc., along with or instead of an actual JSP, [0064])*.

Viswanath teaches the Active Server Page including byte code *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g.,*

*servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])*.

It would have been obvious to one of ordinary skill of the art having the teaching of Goward and Viswanath at the time the invention was made to modify the system of Goward to include the limitations as taught by Viswanath. One of ordinary skill in the art would be motivated to make this combination in order to compile JSP-S into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server in view of Viswanath ([0048]), as doing so would give the added benefit of providing a method and/or system for handling requests from a user and rendering a resulting page in HTML useful in effectuating aspects of an order management system, which reduces the redundancy in the underlying code required to effectuate the page as taught by Viswanath ([0009]).

**As to claims 13, 24**, Goward teaches a method/apparatus comprising:

creating a plurality of user interface templates *(i.e. templates 201-203, Fig. 2; select a static HTML page, a server page or a graphics file to display, [0040])* associated with a Web-based application *(i.e. Application server, Fig. 1)*, wherein the plurality of user interface templates are created using an Active Server Page Language and the Web-based application *(i.e. Application server, Fig. 1)* uses an Internet service application programming interface (ISAPI) *(i.e. main program of Servlet 110, [0048])* to implement business logic separately from the plurality of user interfaces *(i.e. During this execution process, server page 211 can make callbacks into servlet 110, and also to use methods defined within servlet 110, and also to access state information stored within servlet 110, [0045])*;

compiling *(i.e. Server page 211 is first retrieved from its file (step 502), and is compiled into a compiled server page (step 504). Next, the compiled server page is executed to compute any required data values and to generate display page 221 (step 506), [0044])* the plurality of user interface templates into a plurality of byte codes prior to execution *(i.e. compile server page, Step 504, Fig. 5)*; and

storing the plurality of byte codes *(i.e. the server page includes dynamically executable code embedded in a display language, [0011])* associated with the plurality of user interface templates in a single file *(i.e. compile server page, Step 504, Fig. 5)*, wherein the byte codes are capable of being executed by an execution engine in a Web server, the execution *(i.e. execute complied server page to compute data and generate page, Step 506, Fig. 5; [0036; 0043-0050])* engine comprises run time code of the ISAPI that executes the single file derived from the plurality of user interface templates created using an Active Server Page Language to generate Web pages using a system language of the ISAPI *(i.e. the server page includes an Active Server Page (ASP), [0019], During this execution process, server page 211 can make callbacks into servlet 110, and also to use methods defined within servlet 110, and also to access state information stored within servlet 110, [0045])*.

Goward implicitly teaches Active Server Page Language, [0019], [0048] *(i.e. server page 211 includes HTML markup text*.

Goward implicitly teaches byte codes as executable code *(i.e. the server page includes dynamically executable code embedded in a display language, [0011])*.

Goward does not explicitly teach "Active Server Page Language", "byte codes".

Viswanath teaches the Active Server Page *(i.e. Active Server Pages (ASP) by Microsoft, Inc., along with or instead of an actual JSP, [0064])*.

Viswanath teaches the Active Server Page including byte code *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])*.

It would have been obvious to one of ordinary skill of the art having the teaching of Goward and Viswanath at the time the invention was made to modify the system of Goward to include the limitations as taught by Viswanath. One of ordinary skill in the art would be motivated to make this combination in order to compile JSP-S into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server in view of Viswanath ([0048]), as doing so would give the added benefit of providing a method and/or system for handling requests from a user and rendering a resulting page in HTML useful in effectuating aspects of an order management system, which reduces the redundancy in the underlying code required to effectuate the page as taught by Viswanath ([0009]).

**As to claims 2, 19**, Viswanath teaches the user interface template has been complied into a byte code format and the Active Server Page contains the byte code *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide*

*additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])*.

**As per claim 3**, Goward teaches a method as recited in claim 1 wherein the user interface template contains HTML code *(i.e. server page 211 includes HTML markup text, [0048])*.

**As per claim 4,** Viswanath teaches the user interface template contains logic related to displaying information *(i.e. Some portions of the detailed descriptions, which follow, are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed by computer systems. These descriptions and representations are used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like, [0030])*.

**As per claim 5**, Viswanath teaches the Active Server Page includes a plurality of compiled user interface templates *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])*.

**As per claim 6**, Goward teaches one or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 1 *(i.e. FIG. 1 illustrates a collection of servers that operate together in accordance with an embodiment of the present invention. In FIG. 1, a web browser 102 on client 104 communicates across network 106 with a web site 108 on web server 109, [0030])*.

**As to claims 8, 22**, Viswanath teaches the plurality of byte codes includes callback codes that call into the Web-based application code *(i.e. the active render method internally calls render methods in helper classes. The render methods in the helper classes so called populate data in a name value pair. These name value pairs are utilized by Java server pages (JSP's) for generating a new HTML page, one which will effectuate the order operation, accordingly. Advantageously, the render methods in helper classes are re-used in a number of other places, [0015])*.

**As to claims 9, 20**, Viswanath teaches the plurality of byte-codes are executed by an execution in a Web Server *(i.e. Upon completion of processing associated with the corresponding operation of the helper classes, the servlet (or other similarly effective Java application running in a web server or application server providing server-side processing accessing a database and/or performing electronic commerce processing) calls a render method. The render method called correspondingly creates a new page in HTML, based on the processing of the previous screen, by which the user actions were inputted, [0014])*.

**As per claim 10**, Viswanath teaches a method as recited in claim 7 wherein the plurality of byte codes *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])* are contained in an Active Server Page *(i.e. Active Server Pages (ASP) by Microsoft, Inc., along with or instead of an actual JSP, [0064])*.

**As to claims 11, 23**, Viswanath teaches the byte codes include logic related to displaying information *(i.e. Some portions of the detailed descriptions, which follow, are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed by computer systems. These descriptions and representations are used by those skilled in the data processing arts to most effectively convey the substance of*

*their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like, [0030]).*

**As per claim 12**, Viswanath teaches one or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 7 *(i.e. The computer readable and computer executable instructions reside, for example, in data storage features such as computer usable volatile memory 104 and/or computer usable non-volatile memory 106 of FIG. 1, [0033]).*

**As per claim 14**, Viswanath teaches a method as recited in claim 13 further comprising executing the plurality of byte codes when the Web-based application is executed *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048]).*

**As to claims 15, 26**, Viswanath teaches the plurality of byte codes include callback codes that call into the Web-base application code *(i.e. the active render method internally calls render methods in helper classes. The render methods in the helper classes so called populate data in a name value pair. These name value pairs are utilized by Java server pages (JSP's) for generating a new HTML page, one which will effectuate the order operation, accordingly. Advantageously, the render methods in helper classes are re-used in a number of other places, [0015])*.

**As to claims 16, 25**, Viswanath executing a portion of the plurality of byte codes when the Web-based application is executed *(i.e. JSP JSP-S may be compiled into bytecode, e.g., into a servlet, e.g., servlet SV, upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing, [0065]; JSP JSP-S is compiled into bytecode, e.g., into a servlet (e.g., servlet SV) upon recognition by server 100S. Importantly, JSP JSP-S can also call helper classes H, also deployed by server 100S, to provide additional processing. [0048])*.

**As per claim 17**, Viswanath teaches one or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 13 *(i.e. The computer readable and computer executable instructions reside, for example, in data storage features such as computer usable volatile memory 104 and/or computer usable non-volatile memory 106 of FIG. 1, [0033])*.

### *Response to Arguments*

With respect to claims 1-26, Applicants have amended the independent claim 1, 7, 13, 18, 21, 24 to recite a new limitation "when compiled is executed through an application programming interface developed using a system language to generate the requested Web Page in the system language from the user interface template created using the Active Server Page Language"; however, upon further consideration, a new ground(s) of rejection is made in view of newly found prior art.

### *Conclusion*

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham, can be reached on (571) 272-7079. The fax number to this Art Unit is (571)-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Miranda Le/
Primary Examiner, Art Unit 2167